ZILKA·KOTAB
PC
ZILKA, KOTAB & FEECE™

95 SOUTH MARKET ST., SUITE 420
SAN JOSE, CA 95113

TELEPHONE (408) 971-2573
FAX (408) 971-4660

FAX COVER SHEET

Date:	September 29, 2004	**Phone Number**	**Fax Number**
To:	Appeal Briefs- Patents		(703) 305-0942
From:	Kevin J. Zilka		

Docket No.: NAI1P069/99.074.01

App. No: 09/609,690

Total Number of Pages Being Transmitted, Including Cover Sheet: 31

Message:

Please deliver to the Board of Patent Appeals and Interferences.

Thank you,

Kevin J. Zilka

☐ *Original to follow Via Regular Mail* X *Original will Not be Sent* ☐ *Original will follow Via Overnight Courier*

IF YOU DO NOT RECEIVE ALL PAGES OR IF YOU ENCOUNTER
ANY OTHER DIFFICULTY, PLEASE PHONE Erica
AT (408) 971-2573 AT YOUR EARLIEST CONVENIENCE

Practitioner's Docket No. NAI1P069/99.074.01 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Wu et al.

Application No.: 09/609,690 Group No.: 2157
Filed: 7/5/2000 Examiner: Gold, Avi M.
For: HIGH PERFORMANCE PACKET PROCESSING USING A GENERAL PURPOSE PROCESSOR

Mail Stop Appeal Briefs – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

1. Transmitted herewith is the APPEAL BRIEF in this application, with respect to the Notice of Appeal filed on September 27, 2004.

2. STATUS OF APPLICANT

This application is on behalf of other than a small entity.

CERTIFICATION UNDER 37 C.F.R. §§ 1.8(a) and 1.10*

*(When using Express Mail, the Express Mail label number is **mandatory**; Express Mail certification is optional.)*

I hereby certify that, on the date shown below, this correspondence is being:

MAILING

_ deposited with the United States Postal Service in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

37 C.F.R. § 1.8(a)

_ with sufficient postage as first class mail.

37 C.F.R. § 1.10*

_ as "Express Mail Post Office to Addressee"
Mailing Label No. _____ **(mandatory)**

TRANSMISSION

✓ facsimile transmitted to the Patent and Trademark Office, (703) 305 - 0942.

Date: 9/29/2004

Erica L. Farlow
Signature

Erica L. Farlow
(type or print name of person certifying)

** Only the date of filing (' 1.6) will be the date used in a patent term adjustment calculation, although the date on any certificate of mailing or transmission under ' 1.8 continues to be taken into account in determining timeliness. See ' 1.703(f). Consider "Express Mail Post Office to Addressee" (' 1.10) or facsimile transmission (' 1.6(d)) for the reply to be accorded the earliest possible filing date for patent term adjustment calculations.*

3. FEE FOR FILING APPEAL BRIEF

Pursuant to 37 C.F.R. § 1.17(c), the fee for filing the Appeal Brief is:

other than a small entity $330.00

Appeal Brief fee due $330.00

4. EXTENSION OF TERM

The proceedings herein are for a patent application and the provisions of 37 C.F.R. § 1.136 apply.

Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

5. TOTAL FEE DUE

The total fee due is:

Appeal brief fee	$330.00
Extension fee (if any)	$0.00

TOTAL FEE DUE $330.00

6. PAYMENT OF FEES

The commissioner is authorized to charge deposit account 50-1351 (NAI1P069) in the amount of $330.00. A duplicate of this transmittal is attached.

7. FEE DEFICIENCY

If any additional extension and/or fee is required, and if any additional fee for claims is required, charge Deposit Account No. 50-1351 (Order No. NAI1P069).

Signature of Practitioner

Reg. No.: 41,429
Tel. No.: 408-971-2573
Customer No.: 28875

Kevin J. Zilka
Silicon Valley IP Group, PC
P.O. Box 721120
San Jose, CA 95172-1120
USA

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the application of	)
Wu et al.	) Group Art Unit: 2157
Application No. 09/609,690	) Examiner: Gold, Avi M.
Filed: 07/05/00	) Docket No. NAI1P069_99.074.01
For: HIGH PERFORMANCE PACKET PROCESSING USING A GENERAL PURPOSE PROCESSOR	) Date: September 29, 2004

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals and Interferences

APPELLANT'S BRIEF (37 C.F.R. § 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on September 27, 2004.

The fees required under § 1.17, and any required petition for extension of time for filing this brief and fees therefor, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief contains these items under the following headings, and in the order set forth below (37 C.F.R. § 41.37(c)(i)):

I REAL PARTY IN INTEREST
II RELATED APPEALS AND INTERFERENCES
III STATUS OF CLAIMS
IV STATUS OF AMENDMENTS
V SUMMARY OF CLAIMED SUBJECT MATTER

VI ISSUES

VII ARGUMENTS

VIII APPENDIX OF CLAIMS INVOLVED IN THE APPEAL

IX APPENDIX LISTING ANY EVIDENCE RELIED ON BY THE APPELLANT IN THE APPEAL

The final page of this brief bears the practitioner's signature.

I REAL PARTY IN INTEREST (37 C.F.R. § 41.37(c)(1)(i))

The real party in interest in this appeal is Networks Associates Technology, Inc.

II RELATED APPEALS AND INTERFERENCES (37 C.F.R. § 41.37(c) (1)(ii))

With respect to other prior or pending appeals, interferences, or related judicial proceedings that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no other such appeals, interferences, or related judicial proceedings.

Since no such proceedings exist, no Related Proceedings Appendix is appended hereto.

III STATUS OF CLAIMS (37 C.F.R. § 41.37(c) (1)(iii))

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-18, and 30

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims withdrawn from consideration: 19-29
2. Claims pending: 1-18, and 30
3. Claims allowed: None
4. Claims rejected: 1-18, and 30

C. CLAIMS ON APPEAL

The claims on appeal are: 1-18, and 30

See additional status information in the Appendix of Claims.

IV STATUS OF AMENDMENTS (37 C.F.R. § 41.37(c)(1)(iv))

As to the status of any amendment filed subsequent to final rejection, there are no such amendments after final.

V SUMMARY OF CLAIMED SUBJECT MATTER (37 C.F.R. § 41.37(c)(1)(v))

With respect to a summary of Claim 1 et al., as shown in Figure 1, a packet processing device is provided including a control logic processor (e.g. item 10 of Figure 1) for filtering packets according to a set of stored rules and an arithmetic logic processor (e.g. item 20 of Figure 1) for executing packet processing instructions based on the content of the packet. In use, the control logic processor spawns a new thread for each incoming packet, relieving the arithmetic logic processor of the need to do so. The control logic processor and the arithmetic logic processor are integrated via a thread queue. The control logic processor assigns a policy to each incoming packet. See operation 202 of Figure 2, for example. A policy action table stores one or more policy instructions which may be easily changed to update policies to be implemented. The policy action table maps a virtual packet flow identification code to the physical memory address of an action code and a state block associated to the identification code. The arithmetic logic processor processes a packet based on the stored policy assigned to that packet. Note page 4, line 20 – page 9, line 25, for example.

With respect to a summary of Claim 30 et al., the above summary is incorporated, at least in part, by reference. Further provided is an apparatus that includes a control logic unit coupled to an input and the policy condition table for feeding an arithmetic logic unit, which is in turn coupled to the policy action table and the state block for generating an output. Note, for example, items 10 and 20 of Figure 1, and the accompanying description found on page 3, line 10 – page 4, line 20.

VI ISSUES (37 C.F.R. § 41.37(c)(1)(vi))

Following, under each issue listed, is a concise statement setting forth the corresponding ground of rejection.

Issue # 1: The Examiner has rejected Claims 1-3 and 5-15 under 35 U.S.C. 103(a) as being unpatentable over Abraham et al., U.S. Patent No. 5,983,270, in view of Gleeson, U.S. Patent No. 5,627,829.

Issue # 2: The Examiner has rejected Claim 4 under 35 U.S.C. 103(a) as being unpatentable over Abraham et al., U.S. Patent No. 5,983,270, in view of Gleeson, U.S. Patent No. 5,627,829, and further in view of Sinclair, U.S. Patent No. 6,069,827.

Issue # 3: The Examiner has rejected Claim 16 under 35 U.S.C. 103(a) as being unpatentable over Abraham et al., U.S. Patent No. 5,983,270, in view of Gleeson, U.S. Patent No. 5,627,829, and further in view of Murakami et al., U.S. Patent No. 6,065,065.

Issue # 4: The Examiner has rejected Claim 30 under 35 U.S.C. 103(a) as being unpatentable over Abraham et al., U.S. Patent No. 5,983,270, in view of Gleeson, U.S. Patent No. 5,627,829, and further in view of Murakami et al., U.S. Patent No. 6,065,065.

Issue # 5: The Examiner has rejected Claims 17-18 under 35 U.S.C. 103(a) as being unpatentable over Abraham et al., U.S. Patent No. 5,983,270, in view of Gleeson, U.S. Patent No. 5,627,829, in view of Murakami et al., U.S. Patent No. 6,065,065, and further in view of Scales, U.S. Patent No. 5,761,729.

VII ARGUMENTS (37 C.F.R. § 41.37(c)(1)(vii))

The claims of the groups noted below do not stand or fall together. In the present section, appellant explains why the claims of each group are believed to be separately patentable.

Issue #1:

The Examiner has rejected Claims 1-3 and 5-15 under 35 U.S.C. 103(a) as being unpatentable over Abraham et al., U.S. Patent No. 5,983,270, in view of Gleeson, U.S. Patent No. 5,627,829.

Group #1: Claims 1-3, 5, and 9-15

Specifically, with respect to Claim 1, the Examiner continues to rely on col. 2, lines 31-60, and col. 7 of Abraham to meet appellant's claimed "data bus connecting the addressable memory unit and the first and second data processing units." Abraham, however, merely suggests one data processing unit, and thus inherently fails to suggest appellant's claimed "data bus connecting the addressable memory unit and the first and second data processing units" (emphasis added).

It is also noted that the Examiner relies on the disclosure of a "network" to meet appellant's claimed "data bus." Following are exemplary definitions setting forth the broadest plain and ordinary meaning of such terms, showing that a network simply does not meet appellant's claimed "data bus."

> "Network - A group of two or more computer systems linked together. There are many types of computer networks, including:
> 1 local-area networks (LANs) : The computers are geographically close together (that is, in the same building).
> 2 wide-area networks (WANs) : The computers are farther apart and are connected by telephone lines or radio waves.
> 3 campus-area networks (CANs): The computers are within a limited geographic area, such as a campus or military base.
> 4 metropolitan-area networks MANs): A data network designed for a town or city.

> 5 home-area networks (HANs): A network contained within a user's home that connects a person's digital devices. ... "
> http://www.webopedia.com/TERM/n/network.html
>
> "Bus - A collection of wires through which data is transmitted from one part of a computer to another. You can think of a bus as a highway on which data travels within a computer. When used in reference to personal computers, the term *bus* usually refers to *internal bus*. This is a bus that connects all the internal computer components to the CPU and main memory. There's also an expansion bus that enables expansion boards to access the CPU and memory. ... "
> http://www.webopedia.com/TERM/b/bus.html

Still yet, the Examiner now relies on col. 20, lines 51 - 67; and col. 21, lines 1 - 50 from Gleeson to make a prior art showing of appellant's claimed "second data processing unit adapted to process incoming packets according to one of said plurality of instruction sets after the filtering, based on a thread assigned to the incoming packets by the first data processing unit." Moreover, the Examiner asserts that Gleeson teaches the use of a compression routine run after filtering that is called by the transmit manager thread which is assigned before filtering.

Appellant respectfully disagrees with such assertion. Whether the Examiner's statement above is true or not, appellant's claimed unit that "process[es] incoming packets according to one of said plurality of instruction sets" (emphasis added) is simply not disclosed by Gleeson. Specifically, the compression routine initiated by the manager thread in no way suggests use of one of said plurality of instruction sets, as claimed by appellant.

In the latest response of September 02, 2004, the Examiner argues that 'the compression routine's "plurality of instruction sets" include overwriting the original TP/NP header and data with new information which corresponds to a wireless PDU header.' Applicant respectfully disagrees. Specifically, such operations do not meet applicant's claimed "instructions," let alone applicant's claimed "instruction sets."

More importantly, Gleeson fails to even suggest the foregoing specifically claimed processing that is "based on a thread assigned to the incoming packets by the first data processing unit." First, Gleeson does not even suggest a first data processing unit (in addition to the second).

Moreover, Abraham's filter engine (the first data processing unit, per the Examiner) in no way suggests any sort of selection of a thread to be executed by a second data processing unit, as claimed.

In the latest response of September 02, 2004, it is noted that the Examiner has not responded to the preceding arguments regarding applicant's claimed "based on a thread assigned to the incoming packets by the first data processing unit."

For these reasons, appellant's claim elements of Claim 1 are not met by the Examiner's proposed combination. Further, it would not have been obvious to one of ordinary skill in the art at the time of the invention was made to modify Abraham in view of Gleeson to use a second data processing unit adapted to process incoming packets according to one of said plurality of instruction sets after the filtering, based on a thread assigned to the incoming packets by the first data processing unit.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on appellant's disclosure. *In re Vaeck*,947 F.2d 488, 20 USPQ2d 1438 (Fed.Cir.1991).

Appellant respectfully asserts that at least the first and third elements of the *prima facie* case of obviousness have not been met, for the reasons set forth hereinabove.

Group #2: Claims 6-7

With respect to the present group, the Examiner relies on the following excerpts from Abraham (in addition to col. 7) to make a prior art showing of appellant's claimed "wherein said first data processing unit comprises logic for matching a first incoming packet to a stored first rule and for generating a first thread if the first incoming packet matches said first rule, said first thread identifying the location of one of said at least one data processing policies in said policy action table."

> "In accordance with the present invention, a network management program is provided that manages the communication of data packets between an intranetwork and an internetwork. The intranetwork includes a plurality of computers connected via a communications medium. The internetwork includes a plurality of computers connected by other communications media. An operator of a computer connected to the intranetwork inputs vital information regarding users of computers connected to the intranetwork, mapping information regarding computers connected to the intranetwork, and policies to be applied against those users and computers, using a graphical user interface. The GUI communicates the vital user information, mapping information and policies to a database which stores and organizes the vital user information, mapping information and policies. A filter executive optimizes the policies stored in the database into a set of rules for each user and passes the rules to a filter engine. The filter engine filters all outbound data packets transmitted from the intranetwork to the internetwork and verifies all inbound data packets from the internetwork according to the rules provided by the filter executive.
>
> In accordance with other aspects of the present invention, the filter executive also communicates the mapping information stored in the database to a naming service manager which further updates the mapping information and returns the updated mapping information to the filter executive. Consequently, the filter executive filters the data packets according to the most recent mapping information." (col. 2, lines 31-60)
>
> "FIG. 4 is a block diagram of the component parts of the network management program 80 as distributed among the various computers and servers connected to the LAN 44. The GUI 70 of each administrative computer 54 and the network server 50 communicate the information and policies input by the operators of those computers to the rules and logging database 72 located on the network server 50 via the LAN 44. These policies are stored and processed by the rules and logging database 72, which then passes the user policies along to the filter executive 76 along with mapping information for each user. The filter executive 76 optimizes the policies into a set of rules for each user and passes the rules and user mapping information to the filter engine 78. The filter engine 78 filters all outbound IP packets transmitted from the LAN 44 to the Internet 40 and verifies all inbound IP packets from the Internet 40 according to the rules provided to the filter

> engine 78 by the filter executive 76. As this occurs, the naming services manager 74 provides the filter executive 76 with updated mapping information which the filter executive then passes on to the filter engine 78 so that the filter engine begins and ceases filtering of IP packets dynamically as users log into and out of the LAN 44." (col. 9, lines 43 - 65)

After careful review of such excerpts along with the remaining Abraham reference, it is clear that Abraham fails to disclose, teach or suggest "matching a first incoming packet to a stored first rule and for generating a first thread if the first incoming packet matches said first rule, said first thread identifying the location of one of said at least one data processing policies in said policy action table" (emphasis added). Only appellant teaches and claims the generation of a thread based on a packet/rule match, where the thread identifies the location of a data processing policy in a policy action table.

Again, appellant respectfully asserts that at least the first and third elements of the *prima facie* case of obviousness have not been met, for the reasons set forth hereinabove.

Group #3: Claim 8

With respect to Claim 8, the Examiner relies on col. 5, lines 46-67; col. 6, lines 1-4; and col. 7 from Abraham to make a prior art showing of appellant's claimed "wherein said data processing policy comprises a first address pointer to a starting address of a first set of instructions and a second address pointer to a starting address of a state block stored in said addressable memory unit, said state block used by said first set of instructions for processing the first incoming packet." Appellant respectfully disagrees with this assertion as there is simply no suggestion in Abraham of any sort of data processing policy that includes a first address pointer to a starting address of a first set of instructions and a second address pointer to a starting address of a state block stored in said addressable memory unit, said state block used by said first set of instructions for processing the first incoming packet. Only appellant teaches and claims such a unique data structure/pointer technique.

Moreover, in the latest response mailed September 02, 2004, the Examiner argues that such limitations are met by the following excerpts from Abraham.

> "The network server 50 also includes a processing unit 62, a display 64 and a mass memory 68. The mass memory 68 generally comprises a random access memory (RAM), read only memory (ROM), and a permanent mass storage device, such as a hard disk drive, tape drive, optical drive, floppy disk drive, or combination thereof." (col. 6, lines 55-60)
>
> "The rules and logging database 72 is a relational database stored in mass memory 68 consisting of the tables shown in FIGS. 9A-9D and FIGS. 25A and 25B which are used by the network management program 80 to manage IP packet traffic passing through the network server 50." (col. 7, lines 22-24)

Still yet, the Examiner argues that "[i]t is inherent that the information the system administrator can access would be stored in the mass memory." Whether this alleged assertion is correct or not, the foregoing excerpts, along with the remaining Abraham reference, simply do not even suggest the foregoing specific data structure/pointer technique, as claimed.

Again, appellant respectfully asserts that at least the first and third elements of the *prima facie* case of obviousness have not been met, for the reasons set forth hereinabove.

Issue #3:

The Examiner has rejected Claim 4 under 35 U.S.C. 103(a) as being unpatentable over Abraham et al., U.S. Patent No. 5,983,270, in view of Gleeson, U.S. Patent No. 5,627,829, and further in view of Sinclair, U.S. Patent No. 6,069,827.

Group #1: Claim 4

With respect to Claim 4, the Examiner relies on col. 7; col. 5, lines 46-67; and col. 6, lines 1-4 from Abraham to meet appellant's claimed:

"wherein at least one of said policies comprises:

a first address pointer element for identifying the location in said addressable memory unit of one of said plurality of instruction sets, and

a second address pointer element for identifying the location in said addressable memory unit of a state block."

Specifically, the Examiner argues that Abraham teaches an addressable memory unit of one of a plurality of instruction sets by virtue of database 72 (see Fig. 4 from Abraham below).

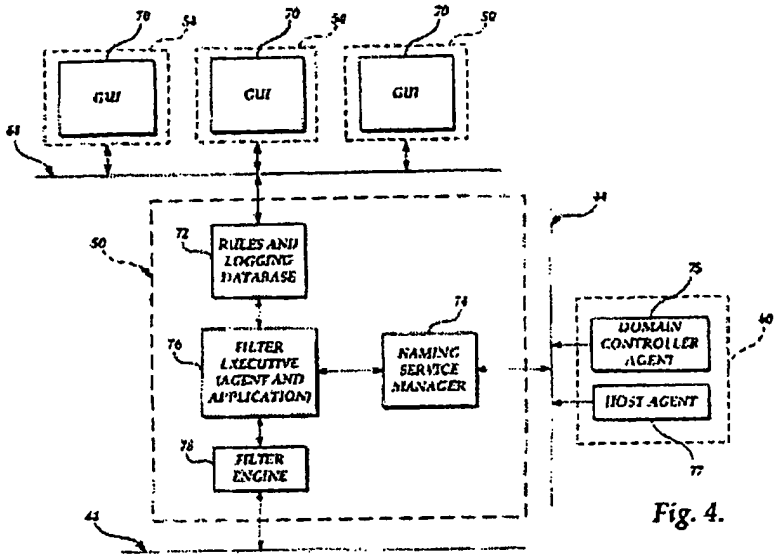

Fig. 4.

The Examiner continues by arguing that col. 5, lines 46-67; and col. 6, lines 1-4 from Abraham discloses appellant's claimed "location in said addressable memory unit of a state block." Such excerpts, however, merely recite the functionality of the domain controller server 60 and client 54 (see above), which are not the database 72.

Thus, Abraham fails to even suggest an addressable memory unit with both instruction sets and a state block, as inherently claimed by appellant. Again, appellant teaches and claims "the location in said addressable memory unit of one of said plurality of instruction sets, and ... the location in said addressable memory unit of a state block." In order to meet appellant's claims,

the database 72 would have to include the claimed state block, which it does not (as implicitly admitted by the Examiner).

Moreover, in the latest response mailed September 02, 2004, the Examiner argues that the limitations of Claim 4 are met by the following excerpts from Abraham.

> "The network server 50 also includes a processing unit 62, a display 64 and a mass memory 68. The mass memory 68 generally comprises a random access memory (RAM), read only memory (ROM), and a permanent mass storage device, such as a hard disk drive, tape drive, optical drive, floppy disk drive, or combination thereof." (col. 6, lines 55-60)
>
> "The rules and logging database 72 is a relational database stored in mass memory 68 consisting of the tables shown in FIGS. 9A-9D and FIGS. 25A and 25B which are used by the network management program 80 to manage IP packet traffic passing through the network server 50." (col. 7, lines 22-24)

Still yet, the Examiner argues that "[i]t is inherent that the information the system administrator can access would be stored in the mass memory." Whether this alleged assertion is correct or not, the foregoing excerpts, along with the remaining Abraham reference, simply do not even suggest the foregoing claimed features.

Again, appellant respectfully asserts that at least the first and third elements of the *prima facie* case of obviousness have not been met, for the reasons set forth hereinabove.

Issue # 3:

The Examiner has rejected Claim 16 under 35 U.S.C. 103(a) as being unpatentable over Abraham et al., U.S. Patent No. 5,983,270, in view of Gleeson, U.S. Patent No. 5,627,829, and further in view of Murakami et al., U.S. Patent No. 6,065,065.

Group #1: Claim 16

This claim is deemed allowable for the reasons base Claim 1 is deemed allowable.

Issue # 4:

The Examiner has rejected Claim 30 under 35 U.S.C. 103(a) as being unpatentable over Abraham et al., U.S. Patent No. 5,983,270, in view of Gleeson, U.S. Patent No. 5,627,829, and further in view of Murakami et al., U.S. Patent No. 6,065,065.

Group #1: Claim 30

With respect to Claim 30, appellant notes numerous deficiencies (including those set forth hereinabove regarding related claims). For example, with respect to appellant's claimed "wherein the apparatus includes a control logic unit coupled to an input and the policy condition table for feeding an arithmetic logic unit, which is in turn coupled to the policy action table and the state block for generating an output," the Examiner relies on the following excerpt from Abraham:

> "FIG. 4 is a block diagram of the component parts of the network management program 80 as distributed among the various computers and servers connected to the LAN 44. The GUI 70 of each administrative computer 54 and the network server 50 communicate the information and policies input by the operators of those computers to the rules and logging database 72 located on the network server 50 via the LAN 44. These policies are stored and processed by the rules and logging database 72, which then passes the user policies along to the filter executive 76 along with mapping information for each user. The filter executive 76 optimizes the policies into a set of rules for each user and passes the rules and user mapping information to the filter engine 78. The filter engine 78 filters all outbound IP packets transmitted from the LAN 44 to the Internet 40 and verifies all inbound IP packets from the Internet 40 according to the rules provided to the filter engine 78 by the filter executive 76. As this occurs, the naming services manager 74 provides the filter executive 76 with updated mapping information which the filter executive then passes on to the filter engine 78 so that the filter engine begins and ceases filtering of IP packets dynamically as users log into and out of the LAN 44." (col. 9, lines 43 - 65)

Such excerpt, however, is replete with deficiencies. Just by way of example, the Examiner states that "Abraham discloses policies stored and processed by a rules and logging database" when addressing appellant's claimed "policy condition table for feeding an arithmetic logic unit." There is simply no arithmetic logic unit in Abraham, let alone a policy condition table feeding the same.

The foregoing excerpt simply fails to even suggest appellant's claimed "control logic unit coupled to an input and the policy condition table for **feeding an arithmetic logic unit, which is in turn coupled to the policy action table and the state block for generating an output**" (emphasis added).

Moreover, in the latest response mailed September 02, 2004, the Examiner argues that "[t]he policy condition table is shown as the set of rules in a database and it is inherent that those policies that are processed would be processed by a logic unit." Whether or not this allegation is true or not, Abraham simply fails to even suggest appellant's claimed "control logic unit coupled to an input and the policy condition table for feeding an arithmetic logic unit, which is in turn coupled to the policy action table and the state block for generating an output."

Again, appellant respectfully asserts that at least the first and third elements of the *prima facie* case of obviousness have not been met, for the reasons set forth hereinabove.

Issue # 5:

The Examiner has rejected Claims 17-18 under 35 U.S.C. 103(a) as being unpatentable over Abraham et al., U.S. Patent No. 5,983,270, in view of Gleeson, U.S. Patent No. 5,627,829, in view of Murakami et al., U.S. Patent No. 6,065,065, and further in view of Scales, U.S. Patent No. 5,761,729.

Group #1: Claims 17-18

These claims are deemed allowable for the reasons base Claim 1 is deemed allowable.

In view of the remarks set forth hereinabove, all of the independent claims are deemed allowable, along with any claims depending therefrom.

VIII APPENDIX OF CLAIMS (37 C.F.R. § 41.37(c)(1)(viii))

The text of the claims involved in the appeal (along with associated status information) is set forth below:

1. (Previously Amended) An apparatus for processing data packets, comprising:

a first data processing unit adapted to filter incoming packets;

an addressable memory unit in which a plurality of instruction sets for packet processing are stored;

a second data processing unit adapted to process incoming packets according to one of said plurality of instruction sets after the filtering, based on a thread assigned to the incoming packets by the first data processing unit; and

a data bus connecting the addressable memory unit and the first and second data processing units.

2. (Original) The apparatus of claim 1, further comprising a policy condition table connected to said first data processing unit, said policy condition table having a plurality of rules stored therein.

3. (Original) The apparatus of claim 1, further comprising a policy action table connected to said data bus and said addressable memory unit, wherein said policy action table stores at least one data processing policy.

4. (Original) The apparatus of claim 3, wherein at least one of said policies comprises:

a first address pointer element for identifying the location in said addressable memory unit of one of said plurality of instruction sets, and

a second address pointer element for identifying the location in said addressable memory unit of a state block.

5. (Original) The apparatus of claim 3, wherein said first data processing unit assigns a thread to each said incoming packet, wherein said thread corresponds to one of said policies stored in said policy action table.

6. (Original) The apparatus of claim 3, wherein said first data processing unit comprises logic for matching a first incoming packet to a stored first rule and for generating a first thread if the first incoming packet matches said first rule, said first thread identifying the location of one of said at least one data processing policies in said policy action table.

7. (Original) The apparatus of claim 6, wherein said second data processing unit is adapted to process the first incoming packet according to said data processing policy corresponding to said first thread.

8. (Original) The apparatus of claim 6, wherein said data processing policy comprises a first address pointer to a starting address of a first set of instructions and a second address pointer to a starting address of a state block stored in said addressable memory unit, said state block used by said first set of instructions for processing the first incoming packet.

9. (Original) The apparatus of claim 6, wherein said thread is assigned to said first incoming packet based on said first rule.

10. (Original) The apparatus of claim 6, wherein said first processing unit further comprises logic for matching a second incoming packet to a stored second rule and for generating a second thread if the second incoming packet matches the second rule, said second thread identifying the location of one of said at least one data processing policy in said policy action table.

11. (Original) The apparatus of claim 10, wherein said second data processing unit is adapted to process the second incoming packet according to said data processing policy corresponding to said second thread.

12. (Original) The apparatus of claim 10, wherein said second thread is assigned to said second incoming packet based on said second rule.

13. (Previously Amended) The apparatus of claim 3, wherein said first processing unit further comprises logic for matching a plurality of incoming packets to a stored corresponding plurality of rules and for generating a thread for each packet that matches one of said plurality of rules, each said thread identifying the location of one of said at least one data processing policy in said policy action table.

14. (Original) The apparatus of claim 13, wherein the second data processing unit is adapted to process each packet according to said data processing policy corresponding to said thread associated with said packet.

15. (Original) The apparatus of claim 13, further comprising a memory unit connected to said first data processing unit and to said second data processing unit, said memory unit adapted to temporarily store packets before processing by said second data processing unit.

16. (Original) The apparatus of claim 1, wherein said second data processing unit comprises a plurality of general purpose processors for executing instructions in parallel.

17. (Original) The apparatus of claim 16, wherein at least one said general purpose processor comprises a complex instruction set computer processor.

18. (Original) The apparatus of claim 16, wherein at least one said general purpose processor comprises a reduced instruction set computer processor.

19. - 29. (Cancelled)

30. (Previously Presented) An apparatus for processing data packets, comprising:

a first data processing unit adapted to filter incoming packets;

an addressable memory unit in which a plurality of instruction sets for packet processing are stored;

a second data processing unit adapted to process incoming packets according to one of said plurality of instruction sets after the filtering, based on a thread assigned to the incoming packets by the first data processing unit; and

a data bus connecting the addressable memory unit and the first and second data processing units;

wherein a policy condition table is connected to said first data processing unit, said policy condition table having a plurality of rules stored therein;

wherein a policy action table is connected to said data bus and said addressable memory unit, wherein said policy action table stores at least one data processing policy;

wherein said first data processing unit comprises logic for matching a first incoming packet to a stored first rule and for generating a first thread if the first incoming packet matches said first rule, said first thread identifying the location of one of said at least one data processing policies in said policy action table;

wherein said second data processing unit is adapted to process the first incoming packet according to said data processing policy corresponding to said first thread;

wherein said data processing policy comprises a first address pointer to a starting address of a first set of instructions and a second address pointer to a starting address of a state block stored in said addressable memory unit,

said state block used by said first set of instructions for processing the first incoming packet;

wherein said first processing unit further comprises logic for matching a second incoming packet to a stored second rule and for generating a second thread if the second incoming packet matches the second rule, said second thread identifying the location of one of said at least one data processing policy in said policy action table;

wherein said second data processing unit is adapted to process the second incoming packet according to said data processing policy corresponding to said second thread;

wherein a memory unit is connected to said first data processing unit and to said second data processing unit, said memory unit adapted to temporarily store packets before processing by said second data processing unit;

wherein said second data processing unit comprises a plurality of general purpose processors for executing instructions in parallel;

wherein the apparatus includes a control logic unit coupled to an input and the policy condition table for feeding an arithmetic logic unit, which is in turn coupled to the policy action table and the state block for generating an output.

IX APPENDIX LISTING ANY EVIDENCE RELIED ON BY THE APPELLANT IN THE APPEAL (37 C.F.R. § 41.37(c)(1)(ix))

There is no such evidence.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 971-2573. For payment of any additional fees due in connection with the filing of this paper, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1351 (Order No. NAI1P069).

Respectfully submitted,

By: _____ Date: 9/29/04

Kevin J. Zilka

Reg. No. 41,429

Zilka-Kotab, P.C.
P.O. Box 721120
San Jose, California 95172-1120
Telephone: (408) 971-2573
Facsimile: (408) 971-4660